

# Responsive Grid System

Responsive Grid System Play+ provides a powerful, responsive grid system based on Bootstrap's 12-column layout. This system enables you to create flexible, mobile-first layouts that adapt seamlessly across different screen sizes.

**Overview** ■ The grid system consists of:

- 12-column layout for maximum flexibility
- 6 responsive breakpoints (xs, sm, md, lg, xl, xxl)
- Container classes for content wrapping
- Column classes for layout control
- Utility classes for advanced positioning

**Breakpoints** ■ Breakpoint Class Prefix Min Width Container Max Width

Breakpoint	Class Prefix	Min Width	Container Max Width
Extra small	None	< 576px	
Auto	Small sm-	≥ 576px	540px
Medium	md-	≥ 768px	720px
Large	lg-	≥ 992px	960px
Extra large	xl-	≥ 1200px	1140px
Extra extra large	xxl-	≥ 1400px	1320px

**Containers** ■ Basic Container ■ `<div class="container">` `<!-- Content with responsive max-width -->` `</div>` Fluid Container ■ `<div class="container-fluid">` `<!-- Content that spans full width -->` `</div>`

**Grid Columns** ■ Basic Column Classes ■ `<div class="row">` `<div class="col-6">` Half width `</div>` `<div class="col-6">` Half width `</div>` `</div>` Responsive Columns ■ `<div class="row">` `<div class="col-12 col-md-6 col-lg-4">` `<!-- Full width on mobile, half on tablet, third on desktop -->` `</div>` `<div class="col-12 col-md-6 col-lg-4">` `<!-- Full width on mobile, half on tablet, third on desktop -->` `</div>` `<div class="col-12 col-lg-4">` `<!-- Full width on mobile/tablet, third on desktop -->` `</div>` `</div>` Auto-sizing Columns ■ `<div class="row">` `<div class="col">` Auto-sized `</div>` `<div class="col">` Auto-sized `</div>` `<div class="col">` Auto-sized `</div>` `</div>`

**Column Widths** ■ Class Width Description

Class	Width	Description
.col-1	8.333%	1/12 of container
.col-2	16.667%	2/12 of container
.col-3	25%	3/12 of container
.col-4	33.333%	4/12 of container
.col-5	41.667%	5/12 of container
.col-6	50%	6/12 of container
.col-7	58.333%	7/12 of container
.col-8	66.667%	8/12 of container
.col-9	75%	9/12 of container
.col-10	83.333%	10/12 of container
.col-11	91.667%	11/12 of container
.col-12	100%	Full width
.col-auto		Auto Content-based width

**Row Columns** ■ Control how many columns appear in a row: `<div class="row row-cols-1 row-cols-md-2 row-cols-lg-3">` `<div class="col">` 1 column per row on mobile `</div>` `<div class="col">` 2 columns per row on tablet `</div>` `<div class="col">` 3 columns per row on desktop `</div>` `</div>`

**Gutters** ■ Default Gutters ■ `<div class="row">` `<div class="col">` With default gutters `</div>` `</div>` No Gutters ■ `<div class="row no-gutters">` `<div class="col">` No spacing between columns `</div>` `</div>` Custom Gutter Sizes ■ `<div class="row g-0">` `<!-- No gutters -->` `<div class="row g-1">` `<!-- 0.25rem gutters -->` `<div class="row g-2">` `<!-- 0.5rem gutters -->` `<div class="row g-3">` `<!-- 1rem gutters -->` `<div class="row g-4">` `<!-- 1.5rem gutters -->` `<div class="row g-5">` `<!-- 3rem gutters -->` `</div>` `</div>` `</div>` `</div>` `</div>` `</div>`

**Directional Gutters** ■ `<div class="row gx-3">` `<!-- Horizontal gutters only -->` `<div class="row gy-2">` `<!-- Vertical gutters only -->` `</div>` `</div>`

**Offsets** ■ Move columns to the right using offset classes: `<div class="row">` `<div class="col-4 offset-4">` Centered column `</div>` `<div class="col-3 offset-3">` Offset by 3 columns `</div>` `</div>`

**Push and Pull** ■ Reposition columns using push and pull classes: `<div class="row">` `<div class="col-9 push-3">`

Pushed to the right </div> <div class="col-3 pull-9"> Pulled to the left </div> </div>

Examples ■ Card Layout ■ <div class="container"> <div class="row row-cols-1

row-cols-md-2 row-cols-lg-3 g-4"> <div class="col"> <div class="card"> <div class="card-body"> <h5 class="card-title"> Card 1 </h5> <p class="card-text"> Responsive card layout </p> </div> </div> </div> <div class="col"> <div class="card"> <div class="card-body"> <h5 class="card-title"> Card 2 </h5> <p class="card-text"> Responsive card layout </p> </div> </div> </div> <div class="col"> <div class="card"> <div class="card-body"> <h5 class="card-title"> Card 3 </h5> <p class="card-text"> Responsive card layout </p> </div> </div> </div> </div> </div> Sidebar Layout ■ <div class="

container-fluid"> <div class="row"> <div class="col-12 col-lg-3"> <aside class="sidebar"> <!-- Sidebar content --> </aside> </div> <div class="col-12 col-lg-9"> <main class="main-content"> <!-- Main content --> </main> </div> </div> </div> Form Layout ■ <div class="container"> <form> <div class="row"> <div class="col-md-6"> <label for="

firstName"> First Name </label> <input type="text" class="form-control" id="firstName"/> </div> <div class="col-md-6"> <label for="lastName"> Last Name </label> <input

type="text" class="form-control" id="lastName"/> </div> </div> <div class="row"> <div class="col-12"> <label for="email"> Email </label> <input type="email" class="

form-control" id="email"/> </div> </div> </form> </div> Best Practices ■ Mobile-first approach : Start with mobile layouts and enhance for larger screens Use semantic HTML :

Combine grid classes with meaningful HTML structure Test across breakpoints : Ensure layouts work well at all screen sizes Avoid nesting too deeply : Keep grid nesting to a minimum for maintainability Use appropriate containers : Choose between .container and .container-fluid based on your design needs CSS Custom Properties ■ The grid system uses CSS custom properties for

gutters: :root { --mdb-gutter-x : 1.5 rem ; /\* Horizontal gutter \*/ --mdb-gutter-y : 0 ; /\* Vertical gutter \*/ } You can customize these values in your theme or override them for specific components.